Rule WLM104: Subsystem (transaction) Service Class did not achieve average response goal

Finding: CPExpert has detected that a service class did not achieve the average response goal that was specified in the Service Policy in effect. This finding applies to performance goals that specify **average response time** as the performance goal. Additionally, this finding applies to service classes that are part of a subsystem (e.g., CICS transactions). This finding is made only if subsystems are installed that support Workload Manager reporting (e.g., at CICS/ESA Version 4.1 or later, and IMS/ESA at Version 5 or later).

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed that support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes that describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system that fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

Notice that the **transactions** comprising the CICSPERS Service Class must actually execute in a CICS region executing CICS at a level of at least CICS/ESA Version 4.1. The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space (e.g., the CICS region) must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class may be one of several "served" transaction service classes controlled by the CICSRGN

Service Class (other CICS transaction service classes "served" by the CICSRGN "server" may be related to procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class).

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICSPERS Service Class, as CICSPERS is simply a logical entity that describes transactions and CICSPERS is not an address space. Rather, the Workload Manager allocates resources to the "server" address space (the CICSRGN Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICSPERS Service Class, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" transaction service classes that are served by the "server" service class. It does this by allocating resources to the "server" service class. **These additional resources may (or may not) be used to provide service to the transaction service class missing its goal**[1].

Suppose there are multiple "served" transaction service classes associated with a "server" service class. If some "served" transaction service class is failing to achieve its goal, the Workload Manager may allocate additional resources to the "server" service class. These additional resources might allow some "served" transaction service classes to significantly exceed their performance goal and these "served" transaction service classes may not be particularly important.

In our example, suppose that the CICSRGN Service Class is serving two transaction service classes (the CICSPERS Service Class we described and a CICSADMN Service Class). Suppose that CICSPERS is important but that CICSADMN is of lower importance. If the Workload Manager detects that CICSPERS is not meeting its performance goal, the Workload Manager may allocate more resources to the CICSRGN Service Class.

[1]Please refer to Section 4 for a more complete illustration of the "server" and "served" concepts.

The CICSRGN would use the additional resources to provide service to both CICSPERS and CICSADMN. Consequently, CICSADMN might significantly exceed its performance goal[2].

To summarize this discussion, performance goals are associated with "served" transaction service classes while resources are allocated to "server" service classes. Performance (i.e., transaction response time) is recorded at the "served" transaction service class level, while resource use is recorded at the "server" service class level.

Subsystem transaction service classes can be defined that have an "average" response goal or a "percentile" response performance goal. An "average" response goal means that the performance goal is defined as transactions should complete within an average of "y" time. A "percentile" response performance goal means that the performance goal is defined as "x%" of the transactions should complete within "y" time. For example, a typical percentile response goal is that **90% of the transactions should complete within 200 milliseconds**.

This rule (Rule WLM104) deals with performance goals for subsystem service classes that have an **average** response goal. Rule WLM105 deals with performance goals for subsystem service classes that have a **percentile** response goal.

The System Resources Manager (SRM) accounts for each transaction executing in the system and determines the transaction's response time[3]. The SRM sums the response times for transactions ending in a service class as each transaction ends. The Workload Manager periodically[4] divides the sum of response times by the number of ending transactions. The result is the average response time of all transactions ending in the service class during the previous interval.

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

[2]Indeed, there is no guarantee that the additional resources would help CICSPERS unless CICSPERS had been properly **defined to CICS** as a higher priority than CICSADMN.

[3]This response time applies only to the time the transaction was in the system; it does not apply to response time delays experienced in the network.

[4]The Workload Manager computes the average transaction response time every 10 seconds, during the "policy evaluation" interval.

For service classes that have an **average response time goal**, the Workload Manager determines whether the average response time achieved by transactions ending in the service class is greater than the performance goal. If the average response time is greater than the performance goal, the system is not meeting performance goals for the service class. If the Goal Importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

At a different interval (typically every 15 minutes), the SRM provides RMF with measurement data, including the elapsed and active times of transactions ending in each service class, and the number of transactions ending in each service class. This information is collected by RMF and written to the SMF data set as Type 72 records. The interval in which RMF collects data and writes records typically is referred to as the *RMF measurement interval*.

RMF does not include in Type 72 records the number of instances in which any service class did not achieve its average response goal. RMF records to total elapsed time and active times and the number of ending transactions.

For response goals, RMF also records in Type 72 records a count of transactions that completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. See Rule WLM102 or Rule WLM105 for a discussion of percentile response performance goals.

The count of transactions completing in varying percentages of the performance goal is useful for analyzing performance of service classes that have a "percentile goal" specified for a service class. However, these counts are not useful in computing average response times.

CPExpert analyzes the SMF Type 72 records to determine whether service class met their performance goals during each RMF measurement interval. For service class that have an average response performance goal specified, CPExpert accomplishes this simply by dividing the number of transactions ending in the service class (R723CRCP) into the elapsed time of ending transactions (R723CTET). The result is the average transaction response time **over the entire RMF measurement interval**.

CPExpert compares the average transaction response time over the entire RMF measurement interval against the performance goal specified for the service class. If the average transaction response time is greater than the performance goal, CPExpert can conclude that the service class did not

achieve its performance goal for the RMF measurement interval. **This conclusion reveals a persistent problem**.

Some transactions executing in the service class may have missed their performance goals, and this situation is to be expected when an average response goal is specified to the Workload Manager. The average response goal simply applies to the *average* response time achieved, which implies that the response time of some transactions may be significantly *less* than the goal and others may be significantly *more* than the goal.

It is important to appreciate that the average response time goal may not be met during a number of Workload Manager policy adjustment intervals. This circumstance may not be detected when CPExpert analyzes RMF data as described above, as the averages are computed based on an entire RMF measurement interval. CPExpert will detect a **persistent** problem, but cannot detect **periodic** problems with average transaction response times being greater than the performance goal[5].

CPExpert produces Rule WLM104 when CPExpert detects that a service class did not meet its average response goal for an entire RMF measurement interval. CPExpert reports the total transactions that ended during the interval, and the average response achieved by the transactions. Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the average transaction response time.

For example, suppose that a 100 millisecond average response time had been specified as the performance goal for a service class period serving CICS transactions. CPExpert might detect that the average response time was 350 milliseconds for transactions in the CICS subsystem service class; the performance goal was missed by 250 milliseconds! CPExpert would report the number of transactions and their average response time.

CPExpert would analyze the causes of delay to CICS transactions and report the primary and secondary causes of delay, **if the information is available**. Some subsystems may not provide detailed information about causes of delay[6]. If this case, CPExpert simply lists "data not available" under the primary and secondary causes of delay column.

[5]The Workload Manager does provide another category of service goal (the Percentile Goal) by which users can specify the percentage of transactions which should achieve their service goals. As mentioned earlier, the Percentile Goal is described in Rule WLM102 and Rule WLM105.

[6]Early releases of IMS Version 5 did not correctly report transaction delays.

The subsystem work manager (e.g., CICS) normally reports the causes of delay to the Workload Manager, using the Workload Management Services macros[7].

CICS reports two separate views of the transactions: the *begin_to_end phase* state and the *execution phase*. IMS reports only *execution phase*.

- **Begin_to_end phase**. The begin_to_end phase starts when CICS has classified the transaction[8]. This action normally is done in a CICS TOR region.

- **Execution phase**. The execution phase starts when either CICS or IMS has started an application task to process the transaction. For CICS, this normally is done in a CICS AOR region. For IMS, this is the IMS Message Processing Region (MPR).

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

In our example of CICS transactions, the CICS subsystem work manager would report transaction delays in the following states for the "served" transaction service class:

- **Active state**. The active state indicates that there was a program executing on behalf of the work request in the "served" service class, from the perspective of the work manager. In the case of a CICS region, this means that a CICS task has been dispatched by CICS to process the transaction.

  However, the active state **does not mean that the task is executing** from the perspective of MVS. It simply means that the task has been dispatched by CICS. Other address spaces with a higher system dispatching priority could preempt the task dispatched by CICS, and these other address spaces could be using the CPU. The situation in which the CICS application task is denied use of the CPU is unknown to CICS.

[7]Please refer to Section 4 (Chapter 2.2) for a description of the interaction between subsystems and the Workload Manager.

[8]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

- **Ready state**. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" transaction service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to process transactions in the "served" service class than were dispatched by CICS.

- **Idle state**. The idle state indicates that there were no work requests (e.g., CICS transactions) ready to run in the service class.

- **Waiting for lock**. The waiting for lock state indicates that some work request (e.g., a CICS task) was waiting for a lock.

- **Waiting for I/O**. The waiting for I/O state indicates that the work manager was waiting for some I/O request on behalf of the "served" service class. This state could be waiting on an actual I/O operation or waiting on some other function related to the I/O request.

- **Waiting for conversation**. The waiting for conversation state indicates that the work manager was waiting for a response in a conversation mode.

- **Waiting for distributed request**. The waiting for distributed request state indicates that some function or data must be routed prior to resumption of the work request.

- **Waiting for session to be established locally**. The waiting for session to be established locally means a wait for a session to be established on the current MVS image.

- **Waiting for session to be established in sysplex**. The waiting for session to be established in sysplex means a wait for a session to be established somewhere in the sysplex.

- **Waiting for session to be established in network**. The waiting for session to be established in network means a wait for a session to be established somewhere in the network.

- **Waiting for timer**. The waiting for timer means that a work request was waiting for expiration of a timer.

- **Waiting for another product**. The waiting for another product means that a work request was waiting for another product to provide some service.

- **Waiting for a new latch**. The waiting for a new latch means that a work request was waiting for a new latch. A latch is a short-duration lock.

- **Waiting for SSL thread**. The waiting for SSL thread means that a work request was waiting for a Secure Sockets Layer thread.

- **Waiting for regular thread**. The waiting for regular thread means that a work request was waiting for a regular thread.

- **Waiting for work table**. The waiting for work table means that a work request was waiting for a work table registration.

- **Waiting for unidentified resource**. The waiting for unidentified resource means that the work request was waiting, but that the work manager could not identify the cause of the wait.

The above causes of delay are analyzed by CPExpert in other rules.

The delays are recorded by RMF from two perspectives: (1) the *begin_to_end phase* of work requests in the service class and (2) the *execution phase* of work requests in the service class. CPExpert can analyze delays to transactions from both perspectives[9].

For SMF Type 72 records related to "server" service class (e.g., a CICS region), RMF records information identifying the service classes served by the server service class. This information is in the "Service Class Served Data Section" of the TYPE 72 records. If CPExpert discovers that a "served" service class did not achieve its performance goal, CPExpert identifies the "server" service classes that serve the service class not achieving its performance goal.

The following example illustrates the output from Rule WLM104:

[9]A CPExpert guidance variable (the **PHASE** variable) in USOURCE(WLMGUIDE) controls which phase CPExpert initially analyzes. Please refer to Section 2 for a discussion of how the PHASE guidance variable may be used to direct CPExpert's analysis and why this guidance may be altered.

```
RULE WLM104:  SERVICE CLASS DID NOT ACHIEVE AVERAGE RESPONSE GOAL

   Service Class CICUSRTX did not achieve its response goal during the
   measurement intervals shown below.  The response goal was 0.090 second
   average response, with an importance level of 2.  CICUSRTX was defined
   as a "served" Service Class (e.g., IMS or CICS transactions).  The
   below causes of delay (if available) were based upon EXECUTION PHASE
   samples.  CICUSRTX was served by CICSRGN.

                            -----LOCAL SYSTEM-----
                             TOTAL    AVERAGE   PERF  PLEX   PRIMARY,SECONDARY
   MEASUREMENT INTERVAL      TRANS   RESPONSE   INDX  PI     CAUSES OF DELAY
   13:07-13:12,21JUN1994    14,307     0.120    1.33  1.33   WAIT I/O(76%),READY(18%)
   13:17-13:22,21JUN1994    14,314     0.181    2.01  2.01   WAIT I/O(62%),READY(32%)
   13:22-13:27,21JUN1994    14,287     0.197    1.9   2.19   WAIT I/O(81%),READY(12%)
```

The information associated with Rule WLM104 is shown based on data collected by the *local system,* which is the system being analyzed for performance purposes.

CPExpert also computes and reports a *sysplex* Performance Index. The WLM maintains both a “sysplex Performance Index” and a “local system Performance Index.” Briefly, the WLM first examines the sysplex Performance Index to determine whether a service class period is missing its performance goal and whether action should be taken. After the sysplex Performance Index is examined at a particular Goal Importance level, the WLM then examines the local system Performance Index. Rule WLM140 explains this WLM logic in more detail, and describes the implications of the WLM logic.

Recall that resources are allocated to "server" service classes, and these "server" service have information relating to resources used and relating to possible delays from a system view. After analyzing the information described above related to the "served" service class missing its performance goal, CPExpert analyzes the "server" service class to identify causes of delay from a system view.

In the example of Rule WLM104, CPExpert detected that the CICSUSRTX service class did not achieve its performance goal. After analyzing the delays from the perspective of CICS, CPExpert will analyze the delays to the server (CICSRGN), from the perspective of the overall system.

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

